

WHAT IS CLAIMED IS:

1 1. A method for preventing overrun of an input data buffer within a program having the
2 input data buffer on a stack data structure, the program executing on a computing system, the
3 method comprising:

4 pushing all arguments to a function onto the stack data structure;

5 pushing a return address onto the stack data structure for use in obtaining the memory
6 address for the instruction to be executed upon completion of the function;

7 pushing onto the stack data structure a security token, the security token comprises a
8 randomly generated data value;

9 allocating memory locations on the stack data structure for use as local variables
10 within the function;

11 completing the instructions within the function;

12 retrieving the security token value from the stack data structure; and

13 if the retrieved security token value is identical to the randomly generated data value,
14 return from the function using the return address stored on the stack data structure.

1 2. The method according to claim 1, wherein the method further comprises aborting the
2 operation of the program if the retrieved security token value is not identical to the randomly
3 generated data value.

1 3. The method according to claim 1, wherein the randomly generated data value is
2 determined using a random number generator once each time the program is executed.

15 wherein the complete function module returns from the function if the retrieved
16 security token is determined to be identical to the randomly generated data value by the test
17 module.

1 8. The apparatus according to claim 7, wherein the complete function module aborts the
2 operation of the program if the retrieved security token is determined not to be identical to
3 the randomly generated data value by the test module.

1 9. The apparatus according to claim 8, wherein the randomly generated data value is
2 determined using a random number generator module once each time the program is
3 executed.

1 10. The apparatus according to claim 9, wherein the random number generator module
2 generates the randomly generated data value using a snapshot of a system clock within the
3 computing system obtained before the program first accepts input data.

1 11. The apparatus according to claim 9, wherein the function comprises a subroutine that
2 does not return a data value.

1 12. The apparatus according to claim 9, wherein the function comprises a subroutine that
2 does returns one or more data values.

1

1 13. A computer program product readable by a computing system and encoding a set of
2 computer instructions for preventing overrun of an input data buffer within a program having

1 18. The method according to claim 13, wherein the function comprises a subroutine that
2 does returns one or more data values.

1 19. A computer program product, according to claim 13, wherein the computer data
2 product comprises a set of computer instructions encoded and stored onto a computer-
3 readable storage medium.

1 20. A computer program product, according to claim 13, wherein the computer data
2 product comprises a set of computer instructions encoded within a carrier wave for
3 transmission between computing systems.

007530" 0795950